**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Attorney Docket No.: **20020019PCT-US**

In re Application of: §
§
**WARDWELL** §
§
Serial No.: **10/529,701** §
§
Filed: **29 MARCH 2005** §
§
For: **METHOD AND SYSTEM FOR COLLATING DATA IN A DISTRIBUTED COMPUTER NETWORK** §
§

Confirmation No.: **2061**

Examiner: **CHANKONG, D.**

Art Unit: **2152**

## APPEAL BRIEF

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notification of Non-Compliant Appeal Brief under 37 C.F.R. § 41.37 dated May 19, 2008, please replace the previous Brief with the present Brief.

No fee or extension of time is believed to be required; however, in the event a fee or extension of time is required, please charge that fee to the BAE Deposit Account No. **19-0130**.

## TABLE OF CONTENTS

## REAL PARTY IN INTEREST

The present application is assigned to BAE Systems, the real party of interest.

## RELATED APPEALS AND INTERFERENCES

No related appeal is presently pending.

## STATUS OF THE CLAIMS

Claims 1-12, which were finally rejected by the Examiner as noted in the Final Office Action dated March 18, 2008 and in the Advisory Action dated April 21, 2008, are being appealed.

## STATUS OF AMENDMENTS

A first Amendment was submitted on December 13, 2007 in reply to a non-final Office Action dated September 14, 2007. A second Amendment was submitted on April 4, 2008 in response to a Final Office Action dated March 18, 2008.

## SUMMARY OF THE CLAIMED SUBJECT MATTER

Claim 1 recites multiple sets of data packets are received from multiple of non-synchronous compute nodes physically separated from each other (page 5, lines 16-18; compute nodes **110A-110N** in Figure **1**). Each of the sets of data packets is provided by one of the non-synchronous compute nodes (page 5, lines 16-18). The data packets are then inserted into a software container (page 7, section I) according to user predetermined rules for determining a logical order for the data packets (page 6, lines 1-13; multi-element queue **114** in Figure **1**). Common groups of the data packets are located within the software container according to the user predetermined rules (page 6, lines 15-28). The software container is protected against incomplete groups of the data packets due to system anomalies or quality of service within the distributed computer network according to a grouping criteria (page 7, lines 1-5). The data packets are outputed in respective logical groups that represent an aggregate packet from the non-synchronous compute nodes after the grouping criteria has been met (page 7, lines 10-13; Figure **2B**).

Claim 5 recites an apparatus having means for receiving multiple sets of data packets (page 5, lines 16-18; multi-element queue **114** in Figure **1**) from multiple of non-synchronous compute nodes physically separated from each other (compute nodes **110A-110N** in Figure **1**). Each of the sets of data packets is provided by one of the non-synchronous compute nodes (page 5, lines 16-18). The apparatus also includes means for inserting data packets into a software container (page 7, section I) according to user predetermined rules for determining a logical order for the data packets (page 6, lines 1-13; multi-element queue **114** in Figure **1**). The apparatus further includes means for locating common groups of the data packets within the software container according to the user predetermined rules (page 6, lines 15-28). The software container is protected against incomplete groups of the data packets due to system anomalies or quality of service within the distributed computer network according to a grouping criteria (page 7, lines 1-5). The apparatus has means for outputing the data packets in respective logical groups that represent an aggregate packet from the non-synchronous compute nodes after the grouping criteria has been met (page 7, lines 10-13; Figure **2B**).

Claim 9 recites a recordable type medium having a computer program product for collating data in a distributed computer network having non-synchronous compute nodes. Initially, multiple sets of data packets are received from multiple of non-synchronous compute nodes physically separated from each other (page 5, lines 16-18; compute nodes **110A-110N** in Figure **1**). Each of the sets of data packets is provided by one of the non-synchronous compute nodes (page 5, lines 16-18). The data packets are then inserted into a software container (page 7, section I) according to user predetermined rules for determining a logical order for the data packets (page 6, lines 1-13; multi-element queue **114** in Figure **1**). Common groups of the data packets are located within the software container according to the user predetermined rules (page 6, lines 15-28). The software container is protected against incomplete groups of the data packets due to system anomalies or quality of service within the distributed computer network according to a grouping criteria (page 7, lines 1-5). The data packets are outputed in respective logical groups that represent an aggregate packet from the non-synchronous compute nodes after the grouping criteria has been met (page 7, lines 10-13; Figure **2B**).

## GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. The Examiner's rejection of Claims 1, 5 and 9 under 35 U.S.C. § 103(a) as being unpatentable over *Mann et al.* (US 6,957,281); and

2. The Examiner's rejection of Claims 2, 6 and 10 under 35 U.S.C. § 103(a) as being unpatentable over *Mann et al.* (US 6,957,281) in view of *Turner et al.* (US 6,907,041).

## ARGUMENT

The Examiner's rejections of Claims 1-2, 5-6 and 9-10 are not well-founded and should be reversed.

I. The Examiner's interpretations of *Mann*'s teachings are inconsistent

Claim 1 (and similarly Claims 5 and 9) recites a step of "receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other, wherein each of said sets of data packets is provided by one of said non-synchronous compute nodes."

On page 5 of the Final Office Action, the Examiner asserts that the claimed receiving step is disclosed by *Mann* in Figure **1** as item **210**, in col. 1, lines 45-60 and in col. 4, lines 52-60. However, there is no item **210** in Figure **1** of *Mann*. The Examiner was probably referring to I/O controller **110** having a classification-based packet transferring mechanism **120** that is connected to a single host **140**.

In col. 1, lines 48-53, *Mann* teaches that "a host system generally processes each received packet individually, including identifying a session from the received packet and accordingly identifying a corresponding session on the host system to which the received packet belongs." In col. 1, lines 54-58, *Mann* then teaches that "when a data stream is transmitted continuously under a communication session, each received packet, upon arriving at the host, may need to be incorporated in to the existing data stream that constitutes the same session." Since *Mann* discloses only one host system that is capable of receiving packets, *Mann* does not teach or

suggest the claimed step of "receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other, wherein each of said sets of data packets is provided by one of said non-synchronous compute nodes" (emphasis added).

Col. 4, lines 52-54 of *Mann* discloses a "classification-based packet transferring mechanism **120** may access the received packets from the front end of the packet queue **220**." Col. 4, lines 55-59 of *Mann* then discloses "classification-based packet transferring mechanism **120** may dynamically determine a session number from classification purposes from a buffered packet that is immediately accessible in the front of the packet queue **220**." Since *Mann* teaches that packets being received from the front end of packet queue **220** and the packets are then transferred to a host system **140** (*see* Figure **2**), *Mann* does not teach or suggest the step of "receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other" (emphasis added), as claimed.

On page 2 of the Final Office Action, the Examiner asserts that it would have been obvious to one of ordinary skill that the different nodes can be implemented physically separately from each other. As mentioned above, col. 1, lines 45-60 of *Mann* discloses only one host system that is capable of receiving packets; thus, there is no different node to be physically separated from each other. In addition, col. 4, lines 52-60 of *Mann* discloses that packets are received from the front end of a packet queue **220**, which is contrary to the teachings of one host system for receiving packets as disclosed in col. 1, lines 45-60 of *Mann*. As shown in Figure **2** of *Mann*, packet queue **220** is not part of a host system **140**.

On page 3 of the Advisory Action, the Examiner changes the characterization of *Mann*'s host system as "the apparatus that receives the data packets." In other words, the Examiner would want to characterize *Mann*'s host system as the claimed software container because the claimed software container is "the apparatus that receives the data packets" (*see* multi-element queue **114** in Figure **1**). However, such characterization would contradict with the Examiner's characterization of *Mann*'s packet queue **220** as the claimed software container (*see* below).

II. *Mann*'s teachings are not applicable to the claimed inserting step

Claim 1 also recites a step of "inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets."

On page 5 of the Final Office Action, the Examiner asserts that the claimed inserting step is disclosed by *Mann* in col. 3, lines 6-9, 41-48, col. 4, lines 30-51 and col. 5, lines 18-28. Basically, the Examiner has characterized *Mann*'s packet queue **220** as the claimed software container. According to *Mann* in col. 4, lines 30-41, packet queue **220** is implemented in a first-in-first-out (FIFO) manner, which is not "user predetermined rules for determining a logical order," as claimed. But more importantly, FIFO will not work with the claimed invention because different sets of data packets from different non-synchronous compute nodes tend to arrive at different times, and FIFO cannot provide the correct order to synchronize the various data packets. Moreover, if FIFO is being utilized, then it is not necessary to have the claimed step of "locating common groups of said data packets within said software container according to said user predetermined rules" since the order (*i.e.*, first in, first out) has already been defined. Incidentally, this means *Mann* does not teach or suggest the claimed locating step either.

Again, as shown in Figure **2** of *Mann*, packet queue **220** is located before host system **140**, and packets arrived at packet queue **220** before arriving host system **140**. Thus, *Mann*'s teachings are contrary to the claimed steps of "receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes" and then "inserting said data packets into a software container according to user predetermined rules" when packet queue **220** is being characterized as the claimed software container.

Because the claimed invention recites novel features that are not disclosed by *Mann*, the § 103 rejection is improper.

III. Neither *Mann* nor *Turner* teaches or suggests the claimed inserting step

Claim 2 (and similarly Claims 6 and 10) recites a step of "said inserting further includes inserting said data packets into said software container according to individual packet time reference."

On page 6 of the Final Office Action, the Examiner states that the claimed inserting step is not disclosed by *Mann*, but the Examiner then asserts that it is disclosed by *Turner* in col. 3, lines 31-47. Col. 3, lines 46-47 of *Turner* teaches that "each source timestamp value indicating the time at which said packet left the source." However, the teaching of each source timestamp value indicating the time at which said packet left the source is really not related to the claimed step of "inserting said data packets into said software container" that happens to be based on individual packet time reference."

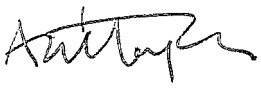
In addition, col. 3, lines 40-42 of *Turner* teaches that "said packets passing from any single source to any single destination are delivered to a single destination in the same order as sent from said single source." Since *Turner* also teaches single source and single destination, it would not be obvious to one skill in the art to modify *Turner*'s teachings (as well as *Mann*'s teachings) to teach the claimed step of "receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other" (emphasis added), as claimed.

Because the cited references, whether separately or combined, do not disclose the claimed invention, the § 103 rejection is improper.

## CONCLUSION

For the reasons stated above, Appellant believes that the claimed invention to be patentably distinct over the cited references, and that the rejections under 35 U.S.C. § 103 are not well-founded. Hence, Appellant respectfully urges the Board to reverse the Examiner's rejection.

Respectfully submitted,



Antony P. Ng
*Registration No. 43,427*
DILLON & YUDELL, LLP
8911 N. Cap. of Texas Hwy., suite 2110
Austin, Texas 78759
(512) 343-6116

ATTORNEY FOR APPELLANT

## CLAIMS APPENDIX

1. A method for collating data in a distributed computer network having non-synchronous compute nodes, said method comprising:

receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other, wherein each of said sets of data packets is provided by one of said non-synchronous compute nodes;

inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets;

locating common groups of said data packets within said software container according to said user predetermined rules;

protecting said software container against incomplete groups of said data packets due to system anomalies or quality of service within said distributed computer network according to a grouping criteria; and

outputting said data packets in respective logical groups that represent an aggregate packet from said non-synchronous compute nodes after said grouping criteria has been met.

2. The method of Claim 1, wherein said inserting further includes inserting said data packets into said software container according to individual packet time reference.

3. The method of Claim 2, wherein said locating further includes locating common groups of said data packets within said software container according to individual packet time reference.

4. The method of Claim 3, wherein said outputting further includes outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

5. An apparatus for collating data in a distributed computer network having non-synchronous compute nodes, said apparatus comprising:

means for receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other, wherein each of said sets of data packets is provided by one of said non-synchronous compute nodes;

means for inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets;

means for locating common groups of said data packets within said software container according to said user predetermined rules;

means for protecting said software container against incomplete groups of said data packets due to system anomalies or quality of service within said distributed computer network according to a grouping criteria; and

means for outputting said data packets in respective logical groups that represent an aggregate packet from said non-synchronous compute nodes after said grouping criteria has been met.

6. The apparatus of Claim 5, wherein said means for inserting further includes means for inserting said data packets into a software container according to individual packet time reference.

7. The apparatus of Claim 6, wherein said means for locating further includes means for locating common groups of said data packets within said container according to individual packet time reference.

8. The apparatus of Claim 7, wherein said means for outputting further includes means for outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

9. A recordable type medium having a computer program product for collating data in a distributed computer network having non-synchronous compute nodes, said recordable type medium comprising:

computer program code for receiving a plurality of sets of data packets from a plurality of non-synchronous compute nodes physically separated from each other, wherein each of said sets of data packets is provided by one of said non-synchronous compute nodes;

computer program code for inserting said data packets into a software container according to user predetermined rules for determining a logical order for said data packets;

computer program code for locating common groups of said data packets within said software container according to said user predetermined rules;

computer program code for protecting said software container against incomplete groups of said data packets due to system anomalies or quality of service within said distributed computer network according to a grouping criteria; and

computer program code for outputting said data packets in respective logical groups that represent an aggregate packet from said non-synchronous compute nodes after said grouping criteria has been met.

10. The recordable type medium of Claim 9, wherein said computer program code for inserting further includes computer program code for inserting said data packets into a software container according to individual packet time reference.

11. The recordable type medium of Claim 10, wherein said computer program code for locating further includes computer program code for locating common groups of said data packets within said container according to individual packet time reference.

12. The recordable type medium of Claim 11, wherein said computer program code for outputting further includes computer program code for outputting said data packets in respective logical groups that represent time-synchronous packets from said non-synchronous compute nodes after said grouping criteria has been met.

## EVIDENCE APPENDIX

Other than the Office Actions and responses already of record, no additional evidence has been entered by Appellant that is relevant to the present appeal.

## RELATED PROCEEDINGS APPENDIX

There is no related proceeding as described by 37 C.F.R. § 41.37(c)(1)(x) known to Appellant, Appellant's legal representative or assignee.